AD-A102 568 CALIFORNIA UNIV BERKELEY CENTER FOR PURE AND APPLIE--ETC F/G 12/1
A FAST SOLVER FREE OF FILL-IN FOR FINITE ELEMENT PROBLEMS,(U)
JUN 81 M R LI, B NOUR-OMID, B N PARLETT N00014-76-C-0013
UNCLASSIFIED PAM-42 NL

1 OF 1
AD A
102568

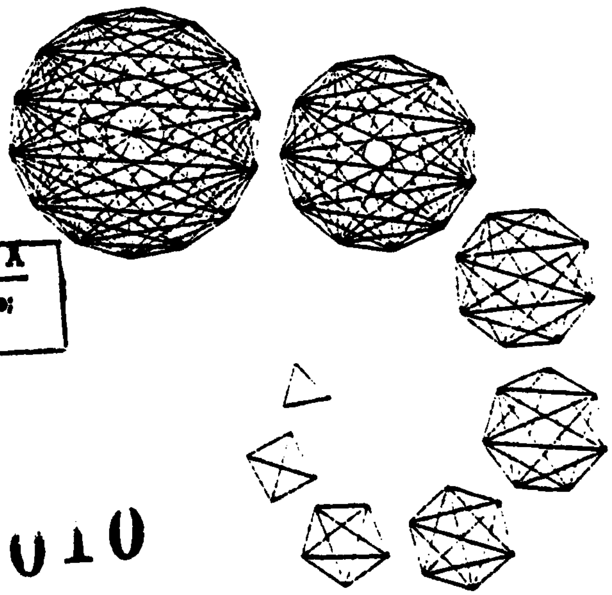
END
DATE
FILMED
9-81
DTIC

CENTER FOR PURE AND APPLIED MATHEMATICS
UNIVERSITY OF CALIFORNIA, BERKELEY

PAM-42

A FAST SOLVER FREE OF FILL-IN FOR FINITE ELEMENT PROBLEMS

M. R. Li, B. Nour-Omit, B. N. Parlett



June 23, 1981

Legal Notice

This report was prepared as an account of work sponsored by the Center for Pure and Applied Mathematics. Neither the Center nor the Department of Mathematics, makes any warranty expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information or process disclosed.

14) TAM-42

(1)

A FAST SOLVER FREE OF FILL-IN FOR FINITE ELEMENT PROBLEMS

by

11)

10) M. R. Li[1] , B. Nour-Omid[2] , B. N. Parlett[3]

12)

1. Beijing Institute of Agricultural Mechanization
2. SESM Division of Civil Engineering Dept., UC Berkeley
3. Math Dept. and Comp. Sci. Division of the EECS Dept., UC Berkeley

The second and third authors gratefully acknowledge partial support by the office of Naval Research Contract 15) N00014-76-C-0013.

404930

ABSTRACT

A new algorithm for solving FEM problems is presented. It blends a preconditioned conjugate gradient iteration into a direct factorization method. The goal was to reduce fill to a negligible level and thus reduce storage requirements but it turned out to be faster than its rivals for an importent class of problems.

1. INTRODUCTION

The numerical method proposed here differs in a single but critical aspect from the one-way dissection algorithm described by Alan George in [George, 1980]. We have been using a dissection procedure for irregular domains which is close to his and so we can simplify our report by focussing on the point of difference and its consequences.

The problem we both address is the solution of an N by N linear system $Ax = b$ arising in the application of finite element methods. In particular we assume that A is symmetric, positive definite, and sparse.

Although nested dissection schemes are, in various ways, asymptotically optimal their preeminence is not apparent for values of N as small as 10^5. Consequently one-way dissection ordering schemes are also worthy of study.

The use of one-way schemes is illustrated by considering block 2 by 2 matrices

$$A = \begin{bmatrix} A_{11} & E^T \\ E & A_{22} \end{bmatrix} \qquad \text{with} \quad A_{ii} = L_i L_i^T, \ i = 1,2.$$

In the course of solving $Ax = b$ by a direct method we must solve, explicitly or implicitly, a reduced system involving the Gauss transform, namely

$$\hat{A}_{22} = A_{22} - E\, A_{11}^{-1}\, E^T \ . \tag{1}$$

Recently George and Liu [Computer Solution of Large Sparse Positive Definite Systems, 1981] have shown a clever way of computing the Cholesky factor $\hat{L}_2$ of $\hat{A}_{22}$ without utilizing any storage beyond E and L_1, the Cholesky factor of A_{11}. Of course $\hat{L}_2$ will be denser than A_{22}.

Our preference however, is to solve the subsystem (1) iteratively, using preconditioned conjugate gradients (CG), without ever forming $\hat{A}_{22}$ explicitly. Our method requires storage for L_1, L_2 and E. Convergence of the iteration is governed by a matrix

$$W = I - C\,C^T \,, \quad \text{where} \quad C = L_2^{-1}\,E\,L_1^{-T} \,,$$

as described in the next section.

There are node orderings in which A_{11} and A_{22} are *dense banded* matrices and in these cases there is no fill at all. On other occasions it pays to allow a little fill in L_1 and L_2, within the band. We follow George in using the word fill rather than fill-in.

Although in general we are prepared to sacrifice some execution time in order to economize on storage we have not had to give up any speed on any of the problems we have tackled so far. On the contrary, our method is significantly quicker than its obvious rivals. Details are in Section 3.

At this point we must say a word about the "sin" of invoking an iteration in the middle of a respectable direct method. We have been so impressed by all the effort and ingenuity which has been expended on reducing fill during

Gaussian elimination that we succumbed to the temptation to get rid of fill either completely, or almost completely.

It will occur to the reader that a more natural way to avoid fill is to solve the original system $Ax = b$ by an iterative technique such as CG. Yet even when the system is preconditioned by the matrix diag (A_{11}, A_{22}) CG is at least twice as slow as our hybrid method.

Of course, a direct method will be quicker when there are many right hand sides. When there are few then the hybrid method is comparable in arithmetic work besides demanding less storage.

The success of our approach depends on the extent to which A_{22} is a good preconditioner for $\hat{A}_{22}$. In Section 2 we analyze the model problem, in part because the analysis can be done quite swiftly with pencil and paper. We show that even simple-minded orderings yield lower arithmetic costs, $O(N^{1.5})$, than comparable direct methods, namely $O(N^{1.75})$ for one-way dissection and $O(N^2)$ for profile solvers. In Section 3 we show the results of several methods on a few realistic problems, including the deflection of a folded plate structure which poses a severe test for our method because of its large condition number. In Section 4 we discuss the efficiency of various methods.

A novel feature of the hybrid approach is that it leads to the search for orderings which keeps A dense around the main diagonal, neither small envelope nor narrow bandwidth are important attributes in this approach.

It has not escpaed our attention that it is possible to use a somewhat more powerful but also more complicated preconditioner than A_{22} for solving $\grave{A}_{22}\ x_2 = c_2$. Both this possibility and the application of these ideas to 3-D problems are under investigation.

2. APPLICATION TO THE MODEL PROBLEM

Consider the 5 point Laplacian operator Δ_h acting on a regular square grid with m^2 unknowns and $h = 1/(m+1)$. There exist very efficient special methods for solving this special problem $-\Delta_h\ u = f$ and the only reasons for considering it here are (1) we can obtain the convergence rate analytically and (2) we get an understanding of how our algorithm behaves.

In order to keep the analysis simple we use a less than optimal ordering in which all the nodes in the odd columns are numbered before the nodes in the even columns, as shown in Fig. 1.

The matrix representation of $-\Delta_h$ which arises from this ordering is shown below.

$$A = \left[\begin{array}{ccc|ccc} D & & & -I & & \\ & \ddots & & -I & \ddots & \\ & & D & & & -I\,-I \\ \hline -I\,-I. & & & D & & \\ & \ddots & & & \ddots & \\ & & -I & & & D \end{array}\right] = \begin{bmatrix} A_{11} & E^T \\ E & A_{22} \end{bmatrix}$$

Here D is a $m \times m$ tridiagonal Toeplitz matrix with nonzero elements $(-1, 4, -1)$. The condition number of A is known,

$$\text{Cond } A = \frac{1 + \cos \pi h}{1 - \cos \pi h} \quad , \quad h = 1/(m+1).$$

$$= \cot^2 (\pi h/2) \doteq 4/\pi^2 h^2 \; , \quad \text{for small } h \; .$$

Let the Cholesky factor of A_{jj} be L_j ; $A_{jj} = L_j L_j^T$, $j = 1,2$. Note that there is no fill when forming L_j . Our algorithm uses the preconditioned conjugate gradient iteration to solve equations with coefficient matrix

$$A_{22} - E\, A_{11}^{-1}\, E^T \; .$$

The matrix which governs the iteration is

$$W = I - (L_2^{-1}\, E\, L_1^{-T})\,(L_1^{-1}\, E^T\, L_2^{-T})$$

and the convergence rate may be bounded in terms of cond (W). Next we determine the largest and smallest eigenvalues of W in the easy case when $A_{11} = A_{22} = H$, m is even, and $L_1 = L_2 = L$.

Note that W is similar to another symmetric matrix $\hat{W} = L\, W\, L^{-1} = I - E\, H^{-1}\, E^T\, H^{-1}$. On writing out $E\, H^{-1}\, E^T\, H^{-1}$ it is seen to be a block tridiagonal Toeplitz matrix with nonzero blocks

$$(\, D^{-2} \, , \, 2D^{-2} \, , \, D^{-2} \,)$$

except for the last block row which is

$$(0, \dots, 0, D^{-2}, D^{-2}).$$

This matrix is a Kronecker (or tensor) product $T \otimes D^{-2}$ where T is the $m/2$ by $m/2$ tridiagonal Toeplitz matrix with nonzero elements $(1, 2, 1)$, except for its last row which is $(0, \dots, 0, 1, 1)$.

The eigenvalues of T and D are well known. See [Gregory and Karney, p. 137] to verify the following, using $h = 1/(m+1)$,

$$\lambda_k (T) = 2\,(1 + \cos 2k\pi h)\ , \quad k = 1, \dots , m/2\ ,$$

$$\lambda_j (D^{-2}) = (4 - 2 \cos j\pi h)^{-2}\ , \quad j = 1, \dots , m\ .$$

The eigenvalues of $T \otimes D^{-2}$ are the $m^2/2$ products of eigenvalues of T and eigenvalues of D^{-2}. Using $\cos m\pi h = -\cos \pi h = 2 \sin^2 \pi h/2 - 1$ we find

$$\lambda_{min} (T \otimes D^{-2}) = 4 \sin^2 \left(\frac{\pi h}{2} \right) \cdot \left[6 \left(1 - \frac{1}{3} \sin^2 \frac{\pi h}{2} \right) \right]^{-2}$$

$$\doteq \frac{1}{36} (\pi h)^2\ , \text{ for small } h\ .$$

$$\lambda_{max} (T \otimes D^{-2}) = 4 \cos^2 (\pi h) \cdot \left[2 \left(1 + 2 \sin^2 \frac{\pi h}{2} \right) \right]^{-2}$$

$$\doteq (1 - \pi^2 h^2)\,(1 - \pi^2 h^2)\ , \text{ for small } h\ .$$

In conclusion

$$\text{cond}\ (W) = \frac{1 - \lambda_{min} (T \otimes D^{-2})}{1 - \lambda_{max} (T \otimes D^{-2})}\ ,$$

$$\doteq \frac{1 - \frac{1}{36} (\pi h)^2}{2(\pi h)^2}\ ,$$

$$\doteq 1/2\ (\pi h)^2\ ,\ \text{for small } h\ .$$

In other words

$$\text{cond}\ (W)\ \doteq\ \text{cond}\ (A)/8\ .$$

Perhaps this last comparison is unfair. If preconditioning is applied to the Gauss Transform matrix $A_{22} - E\,A_{11}^{-1}\,E^T$ it should also be applied to A. It gives the same convergence rate as ordinary CG applied to the matrix

$$\bar{A} = \begin{bmatrix} I & C^T \\ C & I \end{bmatrix}$$

where $C = L^{-1}\,E\,L^{-T}$ and $LL^T = A_{11} = A_{22}$.

To each eigenvalue λ of CC^T , determined above, correspond eigenvalues $\pm\sqrt{\lambda}$ of $\bar{A} - I$. It follows that

$$\text{cond}\ (\bar{A}) = \frac{1 + \sqrt{\lambda_{\max}\ (T \otimes D^{-2})}}{1 - \sqrt{\lambda_{\max}\ (T \otimes D^{-2})}}$$

$$\doteq \frac{2 - (\pi h)^2}{(\pi h)^2} \doteq \frac{2}{(\pi h)^2}\ ,\quad \text{when } h \text{ is small.}$$

Thus, for small h,

$$\text{cond}\ (W) \doteq \text{cond}\ (\bar{A})\ /\ 4 \doteq \text{cond}\ (A)\ /\ 8\ .$$

The condition number yields a crude upper bound on the convergence rate of CG. The standard theory, in [Hestenes and Stiefel, 1952], asserts that when CG is applied in exact arithmetic using a positive definite matrix M then the energy norm of the residual r, i.e., $\sqrt{r^T M^{-1} r}$, is reduced after k steps by at least a factor of

$$2 \left(\frac{\sqrt{\text{cond } (M)} - 1}{\sqrt{\text{cond } (M)} + 1} \right)^k .$$

However for $k >$ order $(M)/2$ this bound is much too big. In our case for small h and modest k the reduction factor per step is at least

$$\begin{aligned} &1 - \pi h &&, \text{ for } A , \\ &1 - \pi h\sqrt{2} &&, \text{ for } \bar{A} , \\ &1 - \pi h 2\sqrt{2} &&, \text{ for } W . \end{aligned}$$

In other words CG with W requires half the number of steps used by CG with $\bar{A}$. One step with W requires less work than one step with $\bar{A}$ and so our hybrid method is more efficient than "pure" preconditioned CG on A for the model problem. Our tests bear this out.

The standard theory given above for estimating convergence is exact only for the Chebyshev - like distribution of eigenvalues of A. In the model problem the eigenvalues are known and more refined bounds can be obtained. For example after k steps the residual norm will have been reduced by at least

$$\frac{(\lambda_N - \lambda_1)\ (\lambda_N - \lambda_2)\ (\lambda_N - \lambda_3)}{(\lambda_1 - 0\ )\ (\lambda_2 - 0\ )\ (\lambda_3 - 0\ )} \cdot \frac{1}{T_{k-3}\left(\frac{\lambda_N + \lambda_4}{\lambda_N - \lambda_4}\right)} ,$$

The eigenvalues λ_i satisfy $0 < \lambda_1 < \lambda_2 < \ldots < \lambda_N$

$$\simeq \frac{1 - 12\,(\pi h)^2}{50\,(\pi h)^2}\,(1 - 4\sqrt{2}\,\pi h)^{k-3} \;.$$

Here T_k is the Chebyshev polynomial of degree k .

Table 1 gives the predicted number of steps S_p and the actual number of steps Sa needed to reduce the residual norm to 10^{-6} of its original value. The refined bounds (using λ_1, λ_2, λ_3, λ_4) are given in parenthesis. Recall that $m + 1 = 1/h$. An unsymmetric load was used for the right hand side.

TABLE 1

$m + 1$	$m = m^2$	Cond (W)	S_p	S_a
20	361	20.79	31 (19)	19
40	1521	81.58	63 (37)	36
80	6241	324.8	124 (77)	66

We turn next to direct methods. These require more storage and have significant advantages when many right hand sides are given. Here we consider only the execution costs for one right hand side.

Our numerical tests showed that the standard profile or envelope schemes, found so often in finite element packages, were significantly slower. The greater the value of N the slower they were relative to N . The reason is simple. Envelope methods require $O(N^2)$ operations On the other hand the hybrid scheme needs $O(N^{1.5})$; each step requires $O(N)$ operations and the number of iterations required to converge to working precision is $O(h^{-1}) = O(m) = O(N^{\frac{1}{2}})$ for the model problem.

Of course, the coefficients of these leading terms are needed to complete the picture. Our tests suggest that for direct and hybrid methods the coefficients are the same order of magnitude. Even for $m = 20$ the hybrid method was twice as fast as the envelope (or profile) Cholesky factorization algorithm.

We have not compared our method with George's One Way Dissection algorithm on the model problem but we expect his to be halfway between ours and the envelope algorithm because George has shown that the number of operations is $O(N^{1.75})$ for the model problem.

We have not given precise comparisons because we do not think that too much weight should be given to the model problem. It is useful in giving insight into the algorithms but it is no benchmark.

In the next section we take up more realistic applications.

3. NUMERICAL EXAMPLES

In this section we compare the hybrid method (called H) described earlier, with the one-way dissection method (called IWD). The test problems used for this comparison are typical of those found in finite element structural analysis. More detailed illustrations of these problems are provided in Figs. 2 and 3. We are aware that a certain method may perform better than all others when applied to a narrow class of problems, and a truer evaluation of the method can only come from a much wider bed of test problems. We will try to point out any bias in our test problems. In any case, they do represent an area where sparse techniques are applied extensively.

In the following tables an 'operation' means a multiplication followed by an addition. Comparisons between execution times were avoided because these times depend very much on the implementations of the algorithms and somewhat on the computer system that is used.

The first set of results was obtained by applying the two methods (IWD & H) to the following:

(i) A plane stress problem producing a matrix of the form shown in Fig. 2. This involves Poisson's equation

$$- \Delta u = f$$

with both Dirichlet and Neumann bounary conditions.

(ii) The fourth order biharmonic equation

$$\Delta^2 u = f$$

inside a rectangular domain with a similar mixed boundary condition. This corresponds to a partially clamped plate problem.

(iii) A folded plate structure is a more practical one and we feel this problem is a crucial test of our method.

These problems have a very large condition number and traditionally iterative methods have not been used because of their depressing rates of convergence.

The domain was discretized using a finite element mesh. Three different node orderings were applied to the mesh and the resulting system of equations was solved, for each ordering, using the above methods. In [2] Alan George derived an expression for the ratio order (A_{11}) : order (A_{22}) . Based on this, we found that the optimum ratio in (i) and (ii) for IWD and H is about 3:1. In (i) and (ii) the right hand side of the equation (load vector) was physically symmetric. This results in a somewhat faster convergence rate than for a general right handside. An unsymmetric load vector increases the number of iterations by about 30%.

In all the test problems the tolerance on the residual norm was set at computer precision and the initial vector was chosen as a zero vector. Of course for a lower tolerance and a better initial vector the number of iterations will be smaller.

Storage

The hybrid method H requires a total of $\frac{1}{2}$ NZA + $(\frac{5}{2} + 4\eta)$N storage cells when conjugate gradient is used for the iterative part. NZA is the total number of non-zero terms in A and η = order (A_{22})/order (A) and for the ordering described in Ref. (1), $0 \leq \eta \leq \frac{1}{2}$. Therefore the hybrid method will require fewer than $\frac{1}{2}$ (NZA + 7N) storage cells. This compares with $\frac{1}{2}$ (NZA + 11 N) for the conjugate gradient method with no preconditioning. Thus there is some justice in saying that H gets by with minimal storage.

Arithmetic Work

Although the number of iterations required to converge is problem dependent (it depends on the eigenvalue distribution of the matrix), the examples presented here demonstrate a substantial saving in operation counts and we feel this will be true for a very large class of finite element problems.

Table (2) - Comparison of the Direct and Hybrid Method for the Plane Stress Problem (N = 220)

η	Direct Method 1WD		Hybrid Method H		
	no. of oper. x 10^3	storage	no. of its.	no. of oper. x 10^3	storage
1/2	702	4982	30	136	2396
1/3	351	3432	23	100	2264
1/4	216	2810	19	84.4	2278
profile	85.8 (50.8)	5382 (4726)	--	--	--

Table (3) - Comparison of the Direct and Hybrid Method for the Plate Problem (N = 330)

η	Direct Method 1WD		Hybrid Method H		
	no. of Oper. x 10^3	storage	no. of its.	no. of oper. x 10^3	storage
1/2	2375	11027	45	425	4890
1/3	1185	7585	35	319	4698
1/4	730	6240	26	245	4779
profile	275 (172)	12110 (10551)	--	--	--

For the profile method the numbers within parenthesis are exact numbers obtained through computations.

Table (4) - Comparison of the Direct and Hybrid Method for the Folded Plate Structure (N = 1109)

η	Direct Method 1WD		Hybrid Method H		
	no. of oper. x 10^6	storage	no. of its.	no. of oper. x 10^6	storage
1/2	90.0	90440	209	11.2	29600
1/5	27.3	44300	130	7.0	29600
profile	14.0	110900	-	-	--

4. EFFICIENCY

The step counts for the model problem, given in Table 1, are greater than they need be. For the sake of an elegant analysis we took the ratio ρ = order (A_{11}) : order (A_{22}) to be 1:1. This ratio depends on the number of separators used in the node ordering phase which is described in [2]. For a reactangular domain a typical choice might designate columns 4, 8, 12, ... as separators with the result that

$$\eta = \frac{\text{order } (A_{22})}{N} = \frac{1}{4}$$

There would be a little fill in the Cholesky factors L_1 and L_2 , whose half-band widths enlarge to 4, but the cost of each step in the iteration is essentially independent of η . Thus the operation count depends on S , the number of steps, which in turn depends on cond (W) . We do not have a formula for cond (W) when $\eta = \frac{1}{3}$ or $\frac{1}{4}$ but empirically we find that cond (W) $\sim \eta$. Thus to compare two schemes, corresponding to η and η' , we use

$$\frac{S}{S'} = \frac{\sqrt{\text{cond }(W)}}{\sqrt{\text{cond }(W)}} = \sqrt{\frac{\eta}{\eta'}}$$

On the model problem the step counts in Table 1 were indeed reduced by factors of $\sqrt{2/3}$ and $\sqrt{2/4}$ when we tried $\eta = 1/3$ and $\eta = 1/4$. What is interesting is that we found the same reduction for the same η values on the 4th order problem of Section 3. We have no explanation for this yet.

Our experience with our hybrid method is limited but in order to sharpen discussion we will make some comparisons, at least for 2-D problems. Profile (or envelope) solvers are very attractive for small problems (say 200 equations) and they can be included in our hybrid program by choosing $\eta = 0$ (no separators). Of course the small operation count is offset by considerable storage demands. For the model problem the hybrid scheme needs $O(N^{1.5})$ operations while profile solvers need $O(N^2)$ operations. The break even point appears to be close to $N = 600$. Consequently we choose this value as the upper limit for small problems.

Our ratings are given in Table 5 which is based on the results, in Table 3, for the partially clamped plate problem.

Table 5. - Rating of Methods for 2-D Problems on Square Domains with Regular Meshes. 1 = Best. NE = number of equations.

	Small N.E. < 600		Medium 600 < N.E. < 10^5		Large 10^5 < N.E.	
	storage	speed	storage	speed	storage	speed
One Way Dissection	2	3	2	3	2	2
Hybrid	1	2	1	1	1	1
Profile	3	1	3	2	3	3

Comments on Table 5

1. For a tree-like domain the hybrid and one way dissection methods win over the profile method much sooner.
2. *Profile solvers are good for long narrow domains.*
3. Hybrid always dominates one-way dissection.

ACKNOWLEDGEMENT

The authors wish to thank Lin sheng Shu for running some of the numerical tests reported here,

REFERENCES

[1] A. George [1980], "An Automatic One-Way Dissection Algorithm for Irregular Finite Element Problems," SIAM Jour. Num. Anal. 17 , No. 6,740-751.

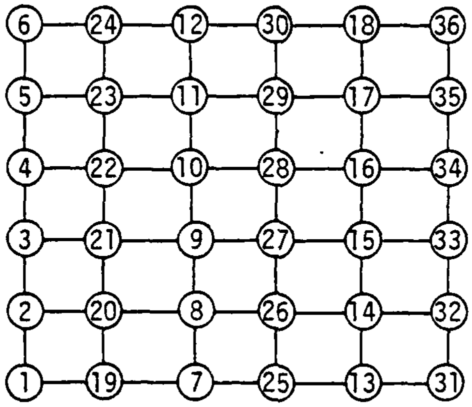
[2] A. George and J. W-H. Liu [1981], *Computer Solution of Large Sparce Positive Definite Systems* , (Prentice Hall, Inc.) .

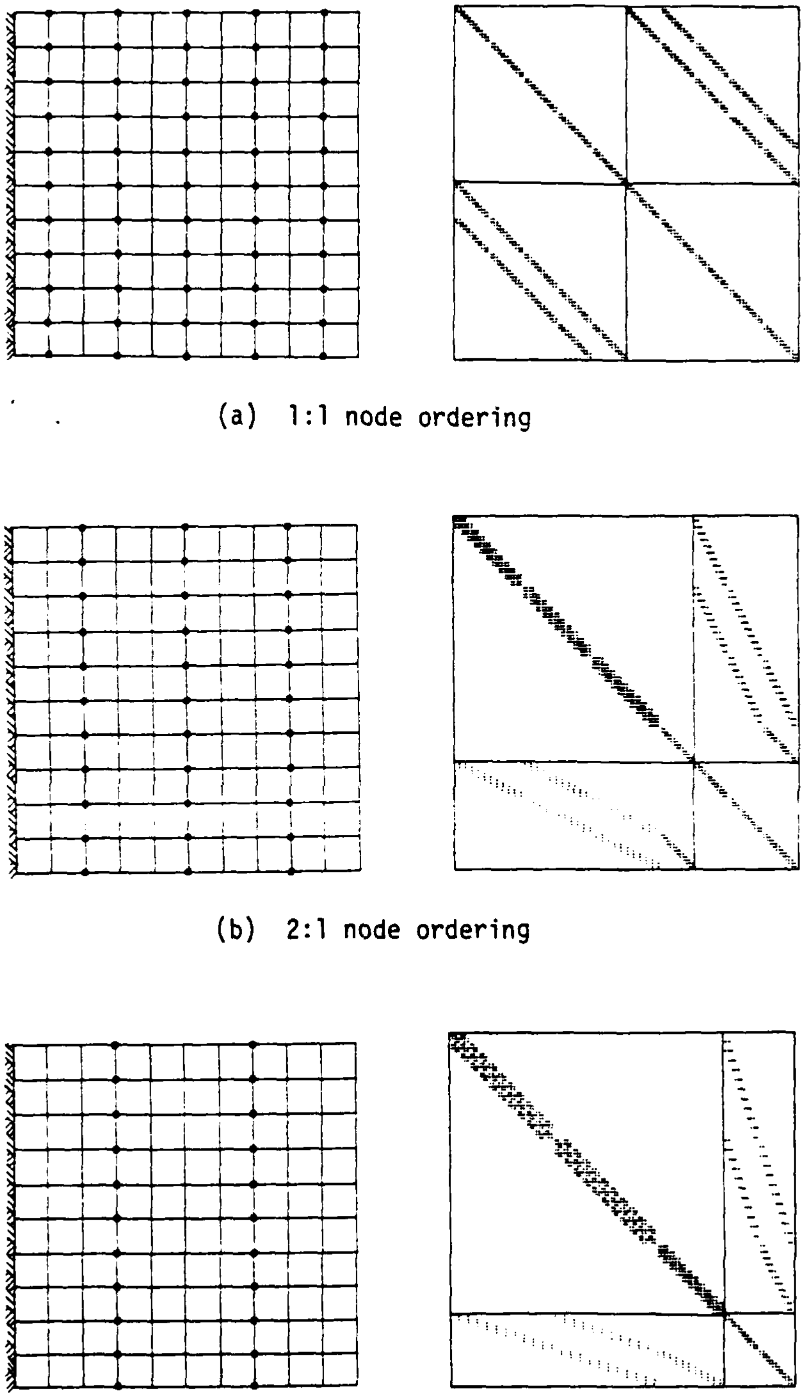
[3] R. T. Gregory and D. L. Karney [1969], *A Collection of Matrices for Testing Computational Algorithms* , (Wily-Interscience, Inc.) .

[4] M. Hestenes and E. Stiefel, " Methods of Conjugate Gradients for solving linear systems," J. Res. Nat. Bur. Standards 49 (1952) .

[5] M. R. Li, [1980], "A Fast and Storage Saving Algorithm for Solving Equations of FEM," Num. Maths., Journal of Chinese Universities, No. 2, 1-11.

FIG. 1


6 24 12 30 18 36
5 23 11 29 17 35
4 22 10 28 16 34
3 21 9 27 15 33
2 20 8 26 14 32
1 19 7 25 13 31



(a) 1:1 node ordering

(b) 2:1 node ordering

(c) 3:1 node ordering

Fig. 2 The Regular Rectangular Mesh Used in the Plane Stress and Plate Problems with Different Node Ordering and the Resulting Matrix Structures.

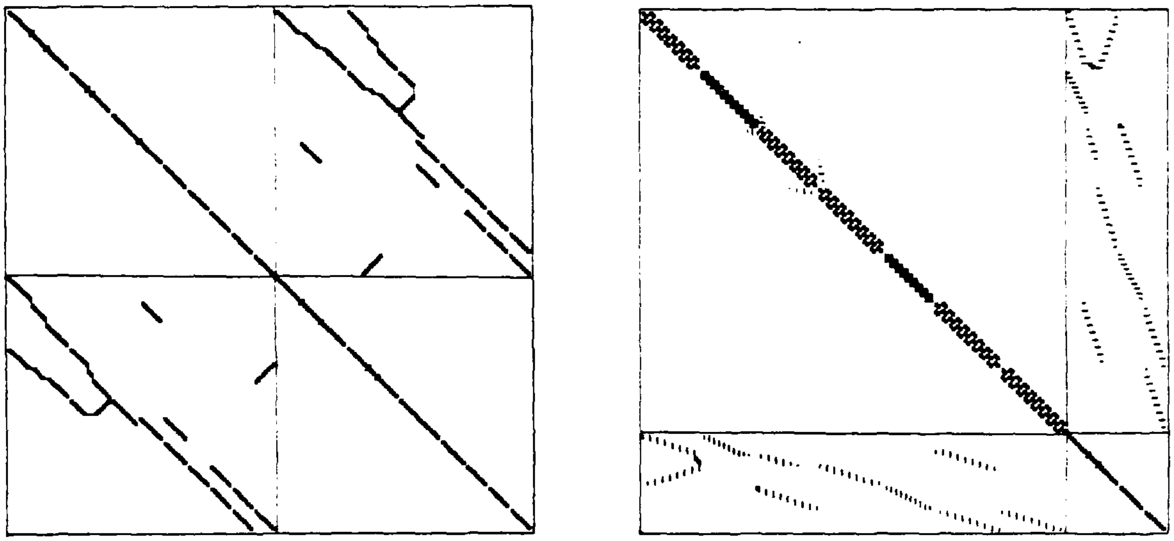


Fig. 3(a) Different Matrix Structures Induced by Two Different Node Orderings of the Folded Plate Structure Below.

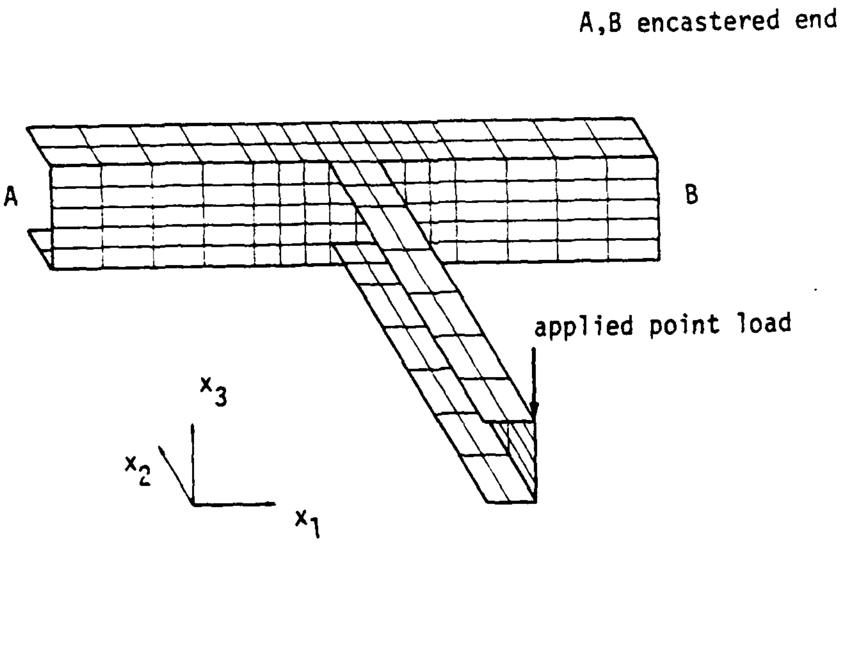


Fig. 3(b) Finite Element Discretization of the Folded Plate Structure.

This report was done with support from the Center for Pure and Applied Mathematics. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Center for Pure and Applied Mathematics or the Department of Mathematics.

DATE

ILMED

-8